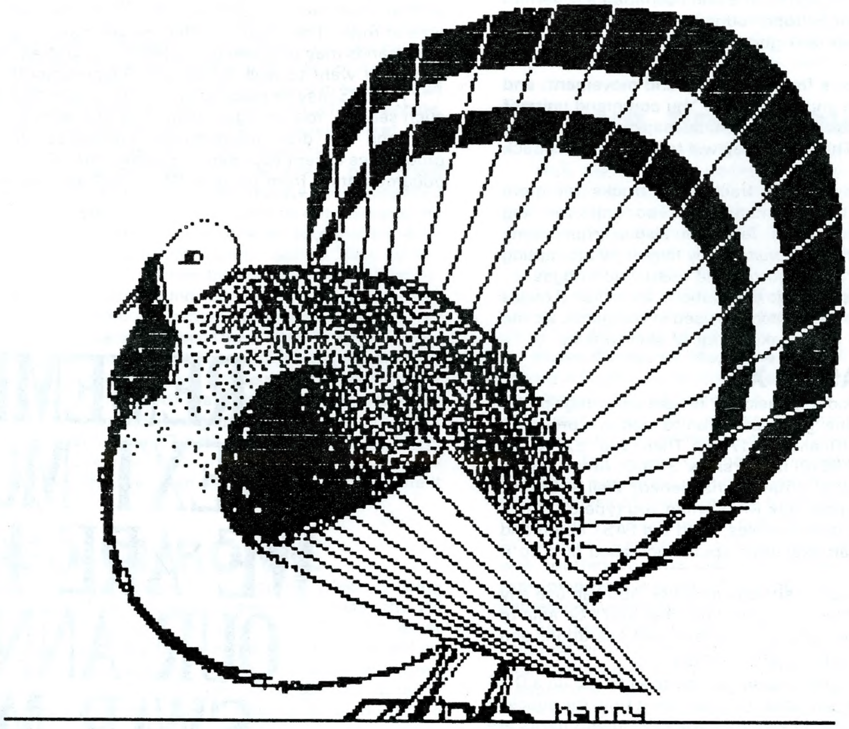
ACE
Atari Computer Enthusiasts

3662 Vine Maple Dr. Eugene OR 97405

NOVEMBER, 1985
Mike Dunn, Jim Bumpas & Larry Gold, Editors



BUMPAS REVIEWS

This is the first issue of the ACE Newsletter in which all the articles are edited using the Atari 520 ST. I'm using ST Writer (Atari Writer ported over) as my word processor. It's really nice having the full 80 columns with which to work. Now I can see better what the article will look like after it's printed.

$10 to The Soft Cellar, 29 Red Cedar Drive, Rochester, NY 14616 will get you "T: A Text Display Device" which is an autoloading machine language utility to display text directly on any graphics screen. An APX award-winner.

WARNING: If you're using DOS2.5XL (the patch to your DOS 2.5 appears in this issue), better keep a copy of your original DOS 2.5. The patch apparently conflicts with the memory location used by SETUP.COM to store the RS232 handler. I tried to setup an autoboot modem program and couldn't get the handler to load when created with DOS 2.5XL.

Otherwise, DOS 2.5XL seems great! It puts the entire DOS-DUP system into the unused area of memory in the lower 64k of the XE/XL machines. With an XE you can then erase the DOS and DUP from the ramdisk and free up the entire 499 sectors for storage. You can still call up DOS at any time with no further disk reads.

PANZER GRENADIER

Panzer Grenadier (SSI, $40) is Roger Damon's latest work. He's still perfecting the game system he developed with Operation Whirlwind and carried through Fire Fight. Five scenarios, each with three levels of difficulty give this novice to intermediate level game (in complexity) enough challenge to keep the experts busy, too.

The scenarios are all WW2 Eastern Front scenarios for one player commanding the Germans against hordes of Russians. The line-of-sight rules for the human player seem more realistic than in his previous games. Once in awhile you get a shot you might not expect. And the Russians seem to be able to fire through any obstruction to get at you.

The maps are plenty large enough for the units commanded. Terrain consists of woods, towns, rivers, hilltops, rough ground and clear. Units include assault guns, tanks, anti-tank guns, mortar, rifle and machine-gun units.

Turns are divided into phases for friendly fire and movement, and separate phases for enemy fire and movement. You command units of company, platoon and section size. The units on the map may be removed at any time to see the terrain. The Option key will toggle the units back onto the map.

Certain foot units can mount half-tracks and trucks for more protection (in the case of half-tracks) and mobility. Foot units can "dig in" for greater protection on the ground. Tanks can also overrun enemy units, and infantry and pioneers can assault enemy targets by attempting to enter their space. Pioneers may also re-build destroyed bridges.

The 8-page manual includes scenario descriptions, as well as 2 pages of drawings and descriptions of the weapons used in the game. At the highest level of difficulty, this game takes a lot of skill to win.

BASIC XE

OSS has expanded their product, Basic XL, to take advantage of the extra memory in the XE machine. It's a lot of fun to load in a program, read the result from PRINT FRE(0) and see 17000. Then type "EXPAND" in direct mode. Now a PRINT FRE(0) might show 31000! And a PRINT FRE(1) might show another 51000! What has happened? Well BASIC XE moves your program into the upper 64k RAM when you type EXPAND. The lower 64k is used for program variables. With the FAST command in your program, OSS claims an execution speed increased by 2 to 6 times.

The product comes on a super-cartridge and disk. You can use any DOS, and if you copy certain files onto your boot disk you'll be able to do such things as renumber your program, perform sort operations and speed up program execution, among other things.

When I first used the product, I automatically tried to call up a DOS menu to see what I had on the disk. Well, you can't do it from the supplied disk. But you don't need to! All, or most, of the DOS menu items are easily and quickly accessible from within BASIC XE. You can load and save binary files. You can read directories, delete, protect and unprotect files. You can rename files.

The manual contains well over 100 pages, including some information reprinted with permission from the Atari Basic Manual. BASIC XE is supposed to be fully compatible with all Atari Basic programs. But I tried to run one without modification which failed when it tried to read string data into memory during the initialization routines. This occured both with the upper 64k in use and without. I haven't gone into the code or the BASIC XE manual to see what there is about these strings which needs changing. BASIC XE supports several string functions not found in other Basics available for the Atari. LEFT$, MID$ and RIGHT$ are supported. FIND is used to determine if a given substring is in a given larger string. HEX$ will convert a decimal number to hex format.

The manual discussions of graphics and player-missiles is much more complete than is found in the Atari Basic Manual. Added Basic statments include PROCEDURE, IF...ELSE, WHILE...ENDWHILE, AND PRINT USING. Discussions are accompanied by many little programming examples. The 5-page index and 5-page table of contents make it possible to find anything you need quickly. If you use an Atari 130 XE, this product will expand your horizons.

GAMES PEOPLE PLAY

Games People Play, 112 East Market St., York, PA 17401, has made a special deal for members of A.C.E. They've given us their 3 disk sides of material plus documentation together with permission to distribute these items to members. Then any A.C.E. member can sign up for their BBS at half price: $15 and receive 5 free hours on the service. They charge $6/hour.

GCP is a graphics BBS for 8-bit Ataris only. I guess if other computers call, they get only text? The disks contain maps of the city and the games (I saw BIOWAR, CYBER SHIP, and CYBER TANK). You can use the joystick to move your personal graphic character down the streets of the town. Go to the postoffice and read mail. Go to a telephone booth and chat with another user. Go into doors for system help, and to go to the games area of town. The town is walled, so you have to go into a door to exit. Commands may be given by keyboard if you're all the way across town and don't want to walk (shades of teleportation!).

The BBS may be reached by a local number through Telenet, or some such service. You configure your boot disk with the local service you'll use, and the disk automatically completes all dialing and log-on procedures except your personal password. After you get the disks and documentation from us, give GCP a call and enjoy!

— Jim Bumpas

**REMEMBER
NEXT MONTH
WE ARE HAVING
OUR ANNUAL
SWAP MEET**

STuff

Jim Bumpas
A.C.E. ST Librarian
4405 Dillard Road
Eugene, OR 97405

We presently have the following items as of October 25, 1985:

ST Writer (an enhanced AtariWriter) — including a lot of support and configuration files — 232k

NEO-Chrome (an excellent drawing program) — 32k

SLIDENEO.PRG and EFFETS.PRG to show *.NEO & *.PIC files — 2k

*.NEO and *.PIC (about 20 files of 32k each)

WINDOW.NEW uses *.PIC files to select pictures for showing (includes *.PIC files) — 332k

Low-Res Demos — 524k

Med-Res Demos — 519k

Hi-Res (monochrome) Demos — 283k

Utilities (ramdisks for half-meg & 1-meg systems) — 210k

If you want our disks, let us know which ones. Until we get a lot of disks, I'll try to cusomize them for you. Send us $15 ($20 for a double-sided disk) and let us know which files you want.

— Jim Bumpas

ST WRITER

ST Writer has some new features not found on the 8-bit Atari Writer. One of the most interesting is the option to receive a file from an 8-bit Atari using the 850 interface. No modem or software is required, but you do need to cable from the 850 into the RS232 port on the ST. Once the Receive option is selected, use DOS to copy the file to the R: device. Or, you can use Atari Writer, or any word processor which will write a file to any device, and write the file to the R: device. Voila!
The file moves at 9600 bps! This is the maximum the 850 interface handles. The computers could do it at 19,000+ bps.

If you use a monochrome monitor you have an option which is really fantastic. You can display double the normal lines of text AND edit in this super mode. Just think: No extra hardware to buy, or special configuration software. And you can display enough text to show nearly a complete 8.5x11" page. The characters are just as clear and easy to read as the larger ones. They're about the same size as normal typing on paper. You'd have to spend more money than it takes to buy an Atari ST system just to give you the same capability on a PC-DOS machine. This machine will make a bigger impact on the business market than any machine since the introduction of PC-DOS machines.

Searches may include carriage returns in the search parameters which allows for much more flexible search and replace operations than I've previously seen in a program. This is a feature I've not seen in word processors even for the IBM. Another nice feature not found on the Atari Writer options menu is the "Transform Colors" option. This switch toggles between black writing on a white background (which I prefer) and white writing on a black background. Color is used in other ways, too. Carriage returns and control characters are red, for instance. It's really exciting to run up the program and see I have 164k for a text buffer! I can put the whole Newsletter in one file! There is a print preview option which permits you to see your text just as it will appear on paper. It might be nice if, as the program is improved, that this feature be integrated into the editing mode. On-screen underlining might also be nice. ST Writer is being distributed with each Atari ST sold now, so if you don't have yours, contact your user group librarian and request a copy. Atari is retaining the copyright, but they've given us permission to distribute it as widely as we desire.

NEO-CHROME

Neo-Chrome is another great Atari Corp. program which is being distributed freely to users. Again, Atari is retaining the copyright. All distribution is with Atari permission. Neo-Chrome v.0.5 is the latest version we have, and it's great! My father-in-law is visiting us and he always enjoys the drawing programs we've had. Two years ago when he was here, all we had was the 8-bit system. He played with it, but it wasn't interesting enough for him to want one for himself. He's a commercial artist. He's buying an Atari ST primarily on the strength of the possibilities he's been shown by the operation of this Neo-Chrome program. That's how good it is. And the promise for the future is even greater.

You can draw on a full screen, or on a half screen with the options menu at the bottom. There is a "grabber" function permitting you to move the picture around so you can work on a critical area and keep the menu on-screen. There are no figure-drawing functions yet (like circles, ellipses, etc. and this is an obvious need in the program). But there are well-implemented copy, cut, paste and move functions. You can use these options to re-locate parts of your picture from one part of the screen to another. You can also mark a part of a picture (up to a half-screen), then load in another picture file and past the old half-screen on top of the new picture. Shades of overlay! And cell drawing! There's a wide selection of brush strokes and another whole mode called "nozzle" with which you can get a simulated "airbrush" effect. Used for shading, you can make it as light or heavy as desired by varying the speed of the mouse as you draw. There's a Text mode with which you can place text in your drawing. You have a half-dozen or so font styles which can each be displayed in one of another half-dozen or so sizes on the screen. There's an "expanded" text mode which seems to invoke the european character set.

One exciting feature permits a kind of simulated "animation". The colors can be shifted (YOU can control the speed of the shift). The well-known waterfall picture demo can now be seen with the water cascading down the falls. Very exciting. Graham Smith is working on a volcano with a lava flow. It should be possible to show moving figures, too, using this feature.

You have a palette of 16 colors with which to work at all times. The menu options permit selection of any one of the 512 colors by placing them into the pots on your palette. One interesting note: These computers can really only display 16 colors at one time. However, the speed of this processor and the screen refresh rate permits one to create the "illusion" of all 512 colors on the screen at one time. The "box" from which colors are selected contains all these hundreds of colors which can be seen at one time. This is a great program with which to begin unleasing the power of the Atari ST. And the price is right! PS: The cover graphic this month was drawn by my father-in-law, Harry Franz. He's a commercial artist, but he's never really cared for computers before. He's seen my 8-bit Atari in the more than 4 years since I've had it. But he's just purchased an Atari ST on the strength of this admittedly unfinished program, NEO-Chrome, which he used to draw the turkey.

News and Reviews

by Mike Dunn, Co-Editor

Well, the rains have come to Oregon, the sun is gone until spring, and the fly-fishing season is over, so back to work on the newsletter. The Atari continues to grow in support, more and more independent reviewers are impressed with the ST, and the STs are selling well. Software is becoming more available for the ST, and much is still coming out for the trusty old 8-bits. Many are converting their 8-bits to 256K and their ST to 1 Megabyte; we will hopefully reprint the directions on how to do it as found in other user group newsletters. See Jim Bumpas' article on the latest in ST news.

Home Computer Magazine (POB 70288, Eugene, OR 97401, $25 year) is a Eugene based computer magazine which started as a TI 99/4 newsletter, and has grown to a huge 146 page tome without any advertising, filled with articles and many programs. Often the same program will be given for the Apple, Commodore 64, IBM-PC, and now the Atari. The current issue has Atari programs like "The NanoProcessor", a program simulating your computer's inner workings, and teaches you the principles of programming a simple machine language computer; "The Plains of Salisbury", a graphics adventure war-game in Camelot; "Vital Signs", a simulation of your cardiovascular system, as well as many well written reviews and the "Programmer's Window" carefully explaining how and why the programs in the newsletter work. Very Impressive, worth getting.

As mentioned in previous newsletters, **PaperClip**, the highly reviewed word-processor from Batteries Included, seemed to have a few strange bugs in it. It lost charactors when you typed very fast, and the cursor sometimes went crazy when deleating and end up in the middle of the page. I have just received a new version, 1.1, which seems to have corrected these problems. All this excellent product needs now is a spelling checker.

Broderbund Software and it's **Synapse** division continues to pour out Atari software at a staggering rate. The last remaining large game software house seems to have faith in us to contiunue to buy their fine products. Released this month, and to be reviewed in this and the next issues of ACE, are included the long-awaited **Championship Lode Runner**, **Karateka** a cinema-like Karate game, two more "Electronic Novels" from Synapse, and more. The reviewers have them now, and they will be printed when received.

VP RAMBLINGS

Here we are into November and turkey day is almost upon us and the Atari scene looks good. More and more software and hardware is being made available to all of us. A number of people have made memory enhancements for all the 8 bit machines. In fact we will have one is use on our BBS in the very near future together with a new 2400 baud modem.

Many of you seem to have trouble either getting on the BBS or once on how to use the different functions. When you go to log on the BBS enter your name, phone no., and a 4 letter password. After hitting the RETURN key a prompt appears to ask if everything is the way you want it and you must answer YES if it is and NO if it is not. If YES then it will check if everything is all right and then let you on the board. If NO you will be allowed to do it over again and only when it gets a YES will it go on and let you on the board. Once on most everything is menu driven and easy to follow except how to send messages. If you just remember these 2 signs you will have no trouble. The first is /A, this one allows you to abort your message. The second is /S and this one sends your message to where you want it. Remember these two prompts and you will have no trouble with the BBS.

The BBS is now working so well that it even scares me. I think we have one of the finest in the country and if you haven't tried it or haven't tried it in a long time I think you will be in for a surprise the next time you use it. If you have any ideas which might make it even better leave me a message and I will see what I can do to follow your idea.

As we go into the holiday season remember to give thanks that we can have computers, modems and all the other things that make computing what it is and enjoy it all in the company of our fellow Atarians.

— Larry Gold

LODE RUNNER'S RESCUE

LODE RUNNER'S RESCUE by Josh Scholar, from Synapse. A challenging 3-D game, with easy to use game editor. My son and his buddys like this game. They had even had more fun, when they found out by using the game editor to win the game.

Now here's the name of the game. The Lode Runner has been captured by the minions of the evil Bungeling Empire and you, his brave daughter, must save him! he's imprisoned in a cell deep within a labyrinth of 46 mazes. Each maze holds many keys, that you must collect while you are avoiding guards. These keys will let you out of each of 46 mazes.

The screen editing Option. This is an icon-driven screen which will let you make your own game screens. The 16-page instruction book covering the game editor is very good. It is easy to understand how to use the icons. This 3D game editor, will let you hide keys behind walls, in pits; to use trapdoors, elevators; to put it in rivers and magic mushrooms; etc. You also can put in restart goals to a new part of the screen if you bite the big one or to catch a magic cat to give you a new life. You can save up to 46 screens per disk, for a game of your own design.

It has a playtest mode so as you make your screens, you can see if it will work. Well that just about covers it for my first review except to say that I liked the game. PS: It's a one player game.

— D.Young

WALDEN'S C

Recently when I was working on an update for DVC, I wanted to know how many bytes a file had in the last sector. I also wanted to know exactly how large the file was. The program FSIZE.C returns the number of bytes, the number of sectors, and how many bytes are in the last sector of the filename. The function inverse() at the end of the program is a handy little function for converting strings to inverse. It is fairly common for C programmers to build up a large library of small functions like this, which can then be included in the LNK file.

Ralph Walden

FLEA MARKET

This year's pre-Christmas flea market to benefit WISTEC will be at our December meeting. If you have anything you wish to sell (original software and/or hardware) contact Larry Gold and give him your $5.00 donation. WISTEC will issue you a receipt for tax purposes.

C-AMIGA

The Commodore AMIGA will appear at our November meeting. Come see the third personal computer entrant in the 68k sweeps (after the Mac and the Atari ST). We'll have the Atari ST there, too, so you can compare the two systems.

1200 XL TO 800 OS

(Reprint: RAM, September 1985)

Bummed out with the hassles built into the 1200XL OS? Wanna burn your translator disks, but can't shell out the $80 for a Boss? Follow these simple instructions to replace the nasty 1200XL OS with the nice old 400/800 OS, for $16.50 or less!

1. Pull the Rev.B ROMs out of the motherboard of a 400, or from the 10k ROM board of an 800. Make sure the chips function properly!

2. If you can't find a set of ROMs, order them from: American TV, 15338 Inverness St., San Leandro, CA 94579, 415-352-3787. They charge $16.50 (including shipping) for the chips.

3. At this point you should have three chips, marked CO12399B, CO12499B and CO14599B.

4. Place the CO14599B piggyback on the CO12499B chip with the notched ends facing the same direction. Solder pin 1 to pin 1, pin 2 to pin 2, etc. for all 24 pairs of pins.

5. Open the 1200XL using a small Phillips-head driver on the screws, and a needlenose pliers to pull the pop rivets holding the RF shield together. This step is completely non-destructive: There is nothing to cut, unsolder or mutilate.

6. Locate the two 24-pin ROM chips on the 1200XL board near the cartridge slot. Notice they are marked U12 and U13 in white letters on the board. Notice also which way the notched ends of the chips are facing.

7. Remove these ROM chips and give them to someone you don't like.

8. Pop the chip marked CO12399B into the slot marked U12, with the notch facing in the proper direction. Place the piggyback chips in slot U13, with the notches properly oriented.

9. Test the board before putting the 1200 together again.

You now have an 800 with a 1200 keyboard, the best of both worlds. This modification was created by Brent Borqhese of Atari Computer Enthusiasts of Columbus, Ohio, courtesy of the Dr. Download BBS, featuring 300/1200 bps access to 3 Mb of public domain programs. Call 614-587-3774, 24 hours daily. No password, no time limit. Just huge phone bills.

Battalion Commander

$40 by Strategic Simulations Inc., 1985

Battalion Commander is an expanded version of Combat Leader, although in many ways not as challenging. The variables which may be altered in this version are fewer, primarily terrain. One of three armies (Soviet - U.S. - Chinese) may be chosen, and one of four campaigns (Pursuit & Exploitation - Meeting Enagement - Attack - Defense). The maps, and there are 40 of them, are twice as large as those in Combat Leader. This gives you quite a lot of territory in which to manuever.

Each game, Attack, Defense etc., has a set number of units, which you may not vary, other than in the relative strengths as presented in a percent ratio. One of the nice things about the game is that each unit is based on the actual figures for a unit of that type in modern U.S., Soviet or Chinese armies. In the back of the instruction manual you will find an appendix which breaks down each unit into title and weapons, both light and heavy. This gives you an understanding of the relative firepower of each unit. A second appendix gives you a breakdown in the number and type of vehicles available for each unit. The third appendix lists the 40 terrain maps and gives a break down of their features.

All of these variables, and the general game mechanics are listed in a nice instruction manual. A typical SSI job, clean and easy to read, with extra bits of interesting information placed at the end of each section. The game mechanics are almost identical to Combat Leader, so anyone familiar with that system should have no trouble.

In play you are allowed to control an individual company or to simply maintain a battalion level control. In either case you can only tell your units how to go to a general area. In each game I've played I have found units trying to move through the enemy lines just because that was the easiest route, taking into account terrain. The only way to avoid this is if you move your units in small increments, and since this is a real time game you are giving the computer a big lead. You may target your units by company unless you have taken over a specific company. Then you may target individual units. I have often given my company commanders fire orders and have had no results, until I took personal command of the company. Some of this may be due to the fact that it is difficult to determine line of sight for your units.

Despite the problems I have just stated, and the fact that I don't find this game to be as challenging as some others I have played by SSI I still recommand any person interested in war games to check this one out. The problems mentioned, while frustrating can also be viewed as a challenge, and in general this game is up to SSI's high standards. It will certainly give you many hours of enjoyable play time.

— Nick Chrones

SORT NUMBER BASIC

FROM LAST ISSUE

```
0 REM ......  FILE: SORTNUM.BAS
10 REM
100 GOSUB 1000:REM WHICH, #, GET DATA
200 GOSUB 2000:REM SORT THE DATA
300 GOSUB 3000:REM PRINT ELAPSED TIME
350 END
400 REM █
             M.L. BUBBLESORT
                                   █
410 REM POKE START/END OF SORT KEY
420 POKE 203,0:POKE 204,4
430 REM REC LENGTH=6, WANT ASCENDING
440 POKE 205,6:POKE 206,0
450 REM NOW MAKE THE USR CALL!!
460 A=USR(ADR(SORT$),ADR(NUM$)+10,N)
470 RETURN
500 K=N
510 T=0
520 FOR I=2 TO K
550 IF S(I-1)>S(I) THEN T=I:H=S(I-1):S
(I-1)=S(I):S(I)=H
570 NEXT I:? ".";
575 K=T-1:IF K>1 THEN 510
580 ? :RETURN
585 REM █
                 HEAPSORT
                                   █
600 R=N
610 FOR LL=INT(N/2) TO 1 STEP -1
620 HOLD=S(LL)
630 GOSUB 680:NEXT LL:LL=1
640 FOR R=N-1 TO 1 STEP -1
650 HOLD=S(R+1)
660 S(R+1)=S(1)
670 GOSUB 680:NEXT R:? :RETURN
680 J=LL
690 I=J:J=2*J
700 ON 2+SGN(J-R) GOTO 710,720,730
710 IF S(J)<S(J+1) THEN J=J+1
720 IF HOLD<S(J) THEN S(I)=S(J):GO TO
690
730 HOLD2=HOLD:HOLD=S(I):S(I)=HOLD2:?
".";:RETURN
780 REM █
             GETKEY ROUTINE
                                   █
800 POKE 764,255
810 IF PEEK(764)=255 THEN 810
820 KEY=PEEK(764)
830 POKE 764,255
840 RETURN
880 REM █
             TIMER ROUTINE
                                   █
900 OLDTIME=TIME
910 FLAG=1-FLAG
920 TIME=PEEK(20)+256*(PEEK(19)+256*PE
EK(18))
930 TIME=INT(1000*TIME/60)/1000
940 ELAPSE=TIME-OLDTIME
950 ? "TIME = ";TIME
960 IF FLAG=0 THEN ? "ELAPSE = ";ELAPS
E
970 RETURN
980 REM █
          INITIALIZATION ROUTINE
                                   █
1000 ? CHR$(125)
1010 POKE 712,0
1020 POKE 710,4+16*INT(16*RND(1))
1030 POKE 709,12
1100 DIM TYP$(1),NUM$(4),S(999),SORT$(
126)
1105 GOSUB 6000
1110 ? :? "Which sort to execute?"
1120 ? :? "(B) Bubblesort"
1125 ? :? "(M) M.L. Bubble"
1130 ? :? "(H) Heapsort":? :?
1140 GOSUB 800
1150 IF KEY=21 THEN TYP$="B"
1155 IF KEY=37 THEN TYP$="M"
1160 IF KEY=57 THEN TYP$="H"
1170 IF TYP$<>"B" AND TYP$<>"M" AND TY
P$<>"H" THEN GO TO 1140
1200 ? :? :? "How many items do you wa
nt to sort"
1205 ? "Max = 999, Minimum = 10"
1210 INPUT NUM$
1220 TRAP 1210
1230 N=VAL(NUM$)
1240 IF N<10 THEN 1205
1250 IF N>999 THEN 1205
1260 N=INT(N):TRAP 34567
1840 OPEN #1,4,0,"D:NUM1000.RAN"
1850 FOR I=1 TO N
1860 INPUT #1,A
1870 S(I)=A
1880 NEXT I
1930 GOSUB 900
1980 RETURN
1990 REM █
                DO SORTS
                                   █
2000 IF TYP$="M" THEN GOSUB 400
2010 IF TYP$="B" THEN GOSUB 500
2020 IF TYP$="H" THEN GOSUB 600
2900 RETURN
2990 REM █
              PRINT RESULTS
                                   █
3000 GOSUB 900
3050 GOSUB 800
3100 FOR I=1 TO N
3150 ? I,S(I)
3200 NEXT I
3900 RETURN
6000 REM USR Sort routine-relocatable
6010 REM Example assumes records in
6020 REM S$, number of records is N.
6030 REM Need to POKE starting and
6040 REM ending positions (relative)
6050 REM of SORT key plus total recrd
6060 REM length, and ascend vs descnd
6070 REM
6080 REM START  SORT KEY:POKE 203,STR
6090 REM END OF SORT KEY:POKE 204,END
6100 REM REC LENGTH=17  :POKE 205,RL
6110 REM ASC=0, DESC=1  :POKE 206,?
6120 REM
6125 ? "          One moment please.."
6130 FOR I=1 TO 126:READ A
6140 SORT$(I)=CHR$(A):NEXT I
6150 RETURN
6160 DATA 104,104,133,217,104,133,216
6170 DATA 104,133,209,104,133,208,169
6180 DATA 0,133,218,133,207,162,1
6190 DATA 165,216,133,214,165,217,133
6200 DATA 215,24,165,214,133,212,101
6210 DATA 205,133,214,165,215,133,213
6220 DATA 105,0,133,215,164,203,165
6230 DATA 206,240,10,177,214,209,212
6240 DATA 144,44,240,12,176,19,177
6250 DATA 214,209,212,144,13,240,2
6260 DATA 176,30,200,196,204,240,227
6270 DATA 176,23,144,223,169,1,133
6280 DATA 218,164,205,136,177,214,72
6290 DATA 177,212,145,214,104,145,212
6300 DATA 192,0,208,241,232,224,0
6310 DATA 208,2,230,207,228,208,208
6320 DATA 172,165,209,197,207,208,166
6330 DATA 165,218,201,0,208,144,96
```

COMPUTER ASSISTED DESIGN

```
2 DIM Q$(1):GOTO 1200
3 DIM K(33,2)
5 DIM CIRC(30,2),OLD(30,2)
6 DIM COLOR$(1),RF(30),VW(30)
7 PI=22/7:ARC=(PI/2)/90
8 GOSUB 9500:GOSUB 9200:RETURN
12 GOSUB 7000
13 C=(RND(0)*14)+1
14 SETCOLOR 2,C,0
15 SETCOLOR 4,C,0
16 SETCOLOR 1,C,8
30 ? CHR$(125)
40 ? "                                     
  "
45 ? " C O M P U T E R  A S S I S T E
D "
46 ? "                                     
  "
47 ? "                D E S I G N         
  "
50 ? "                                     
  "
52 RETURN
100 C=RND(0)*16
110 GRAPHICS 8+16
120 SETCOLOR 1,C,0
130 SETCOLOR 2,C,10
140 SETCOLOR 4,C,10
145 COLOR 1
150 RETURN
160 GOSUB 100
200 FOR L=1 TO HORS
210 DIS=(150/(HORS-1))
220 CIRC(L,1)=180
230 CIRC(L,2)=((DIS*L)-DIS)+21
235 OLD(L,1)=CIRC(L,1)
236 OLD(L,2)=CIRC(L,2)
240 NEXT L
250 RETURN
300 FOR D=1 TO HORS
320 PLOT CIRC(D,1)-40,CIRC(D,2)
330 NEXT D
335 PLOT 160,0:DRAWTO 160,191
336 GOSUB 600
340 PLOT CIRC(1,1),CIRC(1,2)
360 FOR D=2 TO HORS
370 DRAWTO CIRC(D,1),CIRC(D,2)
380 NEXT D
400 IF STRIG(0)=0 THEN GOSUB 600
410 S=STICK(0)
415 IF PEEK(53279)=6 AND VERT>1 AND AN
G>0 THEN GOSUB 7000:GOTO 900
416 IF PEEK(53279)=6 AND ANG<1 THEN 12
00
417 IF PEEK(53279)=6 AND VERT<1 THEN G
RAPHICS 0:GOTO 1200
419 IF PEEK(53279)=5 THEN GRAPHICS 0:G
OTO 1200
420 IF S=15 THEN 400
430 IF S=14 THEN CIRC(PRES,2)=CIRC(PRE
S,2)-1:GOTO 800
440 IF S=13 THEN CIRC(PRES,2)=CIRC(PRE
S,2)+1:GOTO 800
450 IF S=11 THEN CIRC(PRES,1)=CIRC(PRE
S,1)-1:GOTO 800
460 IF S=7 THEN CIRC(PRES,1)=CIRC(PRES
,1)+1:GOTO 800
470 IF S=6 THEN CIRC(PRES,1)=CIRC(PRES
,1)+1:CIRC(PRES,2)=CIRC(PRES,2)-1:GOTO
 800
475 IF S=5 THEN CIRC(PRES,1)=CIRC(PRES
,1)+1:CIRC(PRES,2)=CIRC(PRES,2)+1:GOTO
 800
480 IF S=9 THEN CIRC(PRES,1)=CIRC(PRES
,1)-1:CIRC(PRES,2)=CIRC(PRES,2)+1:GOTO
 800
485 IF S=10 THEN CIRC(PRES,1)=CIRC(PRE
S,1)-1:CIRC(PRES,2)=CIRC(PRES,2)-1:GOT
O 800
499 GOTO 400
600 PRES=PRES+1
610 IF PRES=HORS+1 THEN GOTO 690
620 RUB=PRES-1
630 IF RUB=0 THEN RUB=HORS
640 COLOR 2
650 PLOT 130,0:DRAWTO 130,191
660 COLOR 1
670 PLOT 130,CIRC(PRES,2)
685 RETURN
690 COLOR 2:PLOT 130,0:DRAWTO 130,191:
PRES=1
691 PLOT 140,0:DRAWTO 140,191:COLOR 1
695 FOR T=1 TO HORS
696 PLOT 140,CIRC(T,2):NEXT T:GOTO 620

800 IF CIRC(PRES,2)<=0 THEN GOSUB 9000
:CIRC(PRES,2)=1:GOTO 400
805 IF CIRC(PRES,2)>=191 THEN GOSUB 90
00:CIRC(PRES,2)=190:GOTO 400
810 IF CIRC(PRES,1)>=160 THEN GOSUB 90
00:CIRC(PRES,1)=161:GOTO 400
820 IF CIRC(PRES,1)>=319 THEN GOSUB 90
00:CIRC(PRES,1)=319:GOTO 400
825 IF PRES=1 THEN GOSUB 850:GOTO 890
830 IF PRES=HORS THEN GOSUB 840:GOTO 8
90
835 GOSUB 840:GOSUB 850:GOTO 890
840 COLOR 2:PLOT OLD(PRES,1),OLD(PRES,
2)
841 DRAWTO OLD(PRES-1,1),OLD(PRES-1,2)

844 COLOR 1:PLOT CIRC(PRES,1),CIRC(PRE
S,2)
845 DRAWTO CIRC(PRES-1,1),CIRC(PRES-1,
2):RETURN
850 COLOR 2:PLOT OLD(PRES,1),OLD(PRES,
2)
851 DRAWTO OLD(PRES+1,1),OLD(PRES+1,2)

854 COLOR 1:PLOT CIRC(PRES,1),CIRC(PRE
S,2)
855 DRAWTO CIRC(PRES+1,1),CIRC(PRES+1,
2):RETURN
890 OLD(PRES,1)=CIRC(PRES,1)
895 OLD(PRES,2)=CIRC(PRES,2):GOTO 400
900 GOSUB 100:GOSUB 4000
910 GOSUB 3000
920 FOR T=1 TO HORS
930 ERROR-   PLOT 160+CIRC(T,1)-160█*K
(1,1),VW(T)+RF(T)*K(1,2)
940 FOR D=1 TO 33
950 X=160+(CIRC(T,1)-160)*K(D,1):Y=VW(
T)+RF(T)*K(D,2)
965 IF Y<0 OR Y>191 THEN B=1:GOTO 999
970 IF B=1 THEN PLOT X,Y:B=0
980 DRAWTO X,Y
999 NEXT D:NEXT T
1000 C=(2*PI)/VERT
1005 FOR X=0 TO (2*PI)-C STEP C
1006 SX=SIN(X):CX=COS(X)
1010 FOR J=1 TO HORS-1
1020 RD1A=CIRC(J,1)-160
1030 RD1B=CIRC(J+1,1)-160
1080 X1=160+RD1A*CX:Y1=VW(J)+RF(J)*SX
1090 X2=160+RD1B*CX:Y2=VW(J+1)+RF(J+1)
*SX
1100 IF Y1<0 THEN Y1=0
1110 IF Y1>191 THEN Y1=191
1120 IF Y2<0 THEN Y2=0
1130 IF Y2>191 THEN Y2=191
1140 PLOT X1,Y1
1150 DRAWTO X2,Y2
1180 NEXT J:NEXT X
1185 GOSUB 5000
1190 IF STRIG(0)=1 THEN 1190
1200 GRAPHICS 0:GOSUB 12
1205 ? :? "    Construction lines   U
P  "
1210 ? :? "    Change viewing angle DO
WN "
1220 ? :? "    Go to drawing page   LE
FT "
1230 ? :? "    Begin new design     RI
GHT"
1240 ? :? "    Draw current design  FI
RE "
1242 ? :? "    Help                OPT
ION"
```

COMPUTER ASSISTED DESIGN CON'T

```
1245 ? :? "        USE JOYSTICK TO SE
LECT    ";
1250 S=STICK(0)
1255 IF PEEK(53279)=3 THEN 10000
1260 IF S<>15 THEN 1300
1270 IF STRIG(0)=0 THEN 1440
1280 GOTO 1250
1300 IF S=14 AND HORS>1 THEN GOSUB 12:
GOTO 1350
1310 IF S=13 AND HORS>1 THEN GOSUB 12:
GOTO 1360
1320 IF STICK(0)=11 AND HORS>1 THEN GO
SUB 100:PRES=HORS:GOSUB 660:GOSUB 7000
:GOTO 335
1330 IF S=7 THEN 1400
1340 GOTO 1250
1350 ? :? :? :? "   Number of lines i
n current"
1354 ? :? "     design is ";VERT;"."
1356 ? :? :? :? "     Change to ";:INP
UT VERT:GOTO 1200
1358 GOTO 1200
1360 ? :? :? :? "     Angle of view in
 current"
1362 ? :? "          design is ";ANG;" d
egrees."
1364 ? :? :? :? "   Change to;"
1366 INPUT ANG:GOSUB 3000:GOTO 1200
1400 GOSUB 12:? :? :? :? " Do you wan
t a new design (Y/N)";:INPUT Q$:IF Q$=
"N" THEN 1200
1410 CLR :GOSUB 8:GOSUB 12
1420 ? :? :? " How many construction
circles";:INPUT HORS:CIRCLES=1:GOSUB 2
00
1430 GOTO 1200
1440 IF HORS<1 THEN 1250
1445 IF VERT<1 THEN GOSUB 12:GOTO 1350
1450 IF ANG<1 THEN GOSUB 12:GOTO 1360
1460 IF HORS>1 THEN GOSUB 7000:GOTO 90
0
1470 GOTO 1250
3000 FOR V=1 TO HORS
3005 A=90-ANG
3010 IF CIRC(V,2)<96 THEN H=96-CIRC(V,
2):N=INT(H*SIN(ARC*A)):VW(V)=CIRC(V,2)
+(H-N):GOTO 3100
3020 H=CIRC(V,2)-96:N=INT(H*SIN(ARC*A)
):VW(V)=CIRC(V,2)-(H-N)
3100 NEXT V
3110 RETURN
4000 FOR V=1 TO HORS
4005 G=CIRC(V,1)-160
4010 RF(V)=INT(G*SIN(ARC*ANG))
4020 NEXT V
4030 RETURN
5000 SOUND 0,10,10,14
5010 FOR X=1 TO 7
5020 FOR T=1 TO 3:NEXT T
5030 SOUND 1,20,10,14
5040 FOR T=1 TO 3:NEXT T
5050 SOUND 1,30,10,14
5060 NEXT X
5070 SOUND 0,0,0,0
5075 SOUND 1,0,0,0
5080 RETURN
7000 SOUND 0,10,10,14
7010 SOUND 1,15,10,14
7020 FOR X=1 TO 8:NEXT X
7997 SOUND 0,0,0,0
7998 SOUND 1,0,0,0
7999 RETURN
8000 IF CIRC(PRES,1)>319 THEN CIRC(PRE
S,1)=319:GOSUB 9000:RETURN
8010 IF CIRC(PRES,2)<0 THEN CIRC(PRES,
2)=1:GOSUB 9000:RETURN
8020 IF CIRC(PRES,2)>191 THEN CIRC(PRE
S,2)=191:GOSUB 9000:RETURN
9000 SOUND 0,200,10,14
9010 SOUND 1,200,6,14
9020 FOR X=1 TO 8:NEXT X
9030 SOUND 0,0,0,0
9048 SOUND 1,0,0,0
9050 RETURN
9200 FOR D=1 TO 33
9205 X=X+0.2
9210 K(D,1)=COS(X):K(D,2)=SIN(X)
9220 NEXT D:RETURN
9500 GOSUB 12:X=0
9510 POSITION 6,10:? "(C) SAM SMALL AU
G 1983 (V.2)"
9520 POSITION 8,16:? "  Please wait a
moment  ";
9530 RETURN
10000 GOSUB 12
10010 ? :? "[1]  CLOAD and RUN."
10020 ? "[2]  MENU displayed."
10030 ? "[3]  All functions are locked
 out.     except 'new design'."
10040 ? "[4]  Select 'new design'."
10050 ? "[5]  'New design Y/N' is disp
layed for  checking."
10060 ? "[6]  Program creates COS and
SIN arrays    for faster circle drawin
g"
10070 ? "[7]  Select how many construc
tion          circles are required."
10080 ? "[8]  Return to MENU."
10090 ? "[9]  Drawing page is now avai
lable."
10100 ? :? :? "         PRESS OPTION TO C
ONTINUE  ";
10105 IF PEEK(53279)<>3 THEN 10105
10110 GOSUB 12
10120 ? :? "[10]  DRAWING PAGE"
10130 ? :? "        Select which segment
to be drawn   with fire button"
10140 ? :? "        Joystick will then 'r
ubber band'     drawing line to the d
esired"
10150 ? :? "        profile.  All eight d
irections are   supported and there a
re limit        checks."
10160 ? :? "        START then draws desi
gn if           other parameters have
 been           entered."
10170 ? :? "        SELECT goes back to M
ENU"
10180 ? :? "        PRESS OPTION TO CONT
INUE  ";
10190 IF PEEK(53279)<>3 THEN 10190
10200 GOSUB 12
10210 ? :? "[11]  MAIN MENU"
10220 ? :? "        Attempting to 'draw
current design   without entering oth
er parameters    will result in"
10240 ? :? "        program asking for t
hese entries."
10250 ? :? "        'Draw current design
' will draw      three-dimensional re
presentation     of drawn profile."
10270 ? :? "        PRESS OPTION TO CONT
INUE  ";
10275 IF PEEK(53279)<>3 THEN 10275
10280 GOSUB 12
10290 ? :? :? :? "[12]  PARAMETERS"
10300 ? :? "          'Construction lines
'                 Any number from 1 u
pwards."
10310 ? :? "          'Angle of view'
                  0 deg - eye level,
90 deg - overhead."
10320 POSITION 4,20:? " PRESS OPTION
FOR MAIN MENU  ";
10340 IF PEEK(53279)<>3 THEN 10340
10350 GOTO 1200
```

SECTOR

```
0 REM *****************************
1 REM .  A Disk Utility Programme.
2 REM .    Copyright MAPSOFT Ltd.
4 REM .          By Ron Levy.
5 REM Reprinted from the U.K Atari
6 REM Newsletter, Sussex, England
7 REM by the Ace Newsletter
8 REM 3662 Vine Maple, Eugene, OR
    97405     $14 year
    *****************************
9 BUFF=1536:UNIT=1:POKE 709,15
10 DIM F$(30),OPT$(10),HEX$(512),T$(16
),NVAL$(30),X$(4),Y$(4),SECTOR$(10),DI
SK$(5),B(7),BYT$(256)
12 FOR X=1 TO 5:READ Y:DISK$(X,X)=CHR$
(Y):NEXT X
18 ? "STAGE 1"
19 REM <<< SET UP HEX CONVERTER >>>
20 T$="0123456789ABCDEF"
22 FOR X=1 TO 256
24 Y=INT((X-1)/16):Y2=X-Y*16:Y=Y+1
26 L=LEN(HEX$)+1:HEX$(L,L)=T$(Y,Y)
27 L=LEN(HEX$)+1:HEX$(L,L)=T$(Y2,Y2)
28 NEXT X
30 C=53279:REM Consol Switches.
39 ? "STAGE 2"
40 REM << Set Up Character Array >>
41 FOR BYT=1 TO 255:BYT$(BYT,BYT)=CHR$
(BYT)
44 IF BYT>26 AND BYT<32 OR BYT>124 AND
 BYT<128 OR BYT>154 AND BYT<160 OR BYT
>252 THEN BYT$(BYT,BYT)=CHR$(0)
46 NEXT BYT
50 REM <<<< Create Display List >>>>
52 D=561:D0=PEEK(D):D1=D0-1:DL=PEEK(56
0)+D0*256:DL1=DL-256
53 FOR A=1 TO 6:POKE DL1+A,PEEK(DL+A):
NEXT A
54 FOR A=6 TO 50 STEP 2:POKE DL1+A,0:P
OKE DL1+1+A,2:NEXT A
55 FOR A=52 TO 53:POKE DL1+A,PEEK(DL+A
-23):NEXT A
56 POKE DL1+54,PEEK(561)-1
100 REM <<<<<<<<< Main Menu. >>>>>>>>>
102 GRAPHICS 0:POKE 709,15:CMD=0
105 ? "          Sector Utility."
106 ? "          ================":?
108 ? "            By Ron Levy.":?
112 ? "    Disk Directory ...... <1>"
114 ? "    Load Sector ......... <2>"
116 ? "    Save Sector ......... <3>"
118 ? "    Edit Sector ......... <4>"
120 ? "    Examine Directory ... <5>"
150 POSITION 13,20:? "Option -->";
160 CLOSE #2:OPEN #2,4,0,"K:":GET #2,X
:CLOSE #2
170 X=X-48:IF X<1 OR X>5 THEN 100
180 ON X GOTO 1000,2000,3000,4000,5000
200 FOR I=1 TO SIZE
1000 REM <<<<<< DISK DIRECTORY >>>>>>
1020 GRAPHICS 0:POKE 709,15
1100 TRAP 1600:OPEN #1,6,0,"D:*.*"
1110 INPUT #1,F$:? F$;"  ";
1120 INPUT #1,F$:? F$:GOTO 1110
1600 CLOSE #1:INPUT X$:IF X$="" THEN 1
00
1630 GOTO 1000
2000 REM <<<<<<< Load Sector >>>>>>>
2100 GRAPHICS 0:POKE 709,15:PRINT :TRA
P 40000
2110 ? "       LOAD Sector Routine.":?

2200 ? " Which Sector <HEX> --> ";
2210 INPUT SECTOR$:IF SECTOR$="" THEN
PRINT :GOTO 2400
2220 IF LEN(SECTOR$)<>3 THEN 2000
2230 I=ASC(SECTOR$(1,1)):J=ASC(SECTOR$
(2,2)):K=ASC(SECTOR$(3,3))
2240 IF I<65 THEN I=I-48:GOTO 2260
2250 I=I-55
2260 IF J<65 THEN J=J-48:GOTO 2280
2270 J=J-55
2280 IF K<65 THEN K=K-48:GOTO 2300
2290 K=K-55
2300 SECT=I*256+J*16+K
2310 IF SECT<1 OR SECT>5720 THEN ? CHR
$(253):GOTO 2000
2330 SECTOR=SECT:GOSUB 10000:GOTO 100
2400 ? " Which Sector <DEC> --> ";
2410 INPUT SECTOR$:IF SECTOR$="" THEN
PRINT :GOTO 2500
2430 TRAP 2400:SECT=VAL(SECTOR$):TRAP
40000
2440 IF SECT<1 OR SECT>720 THEN ? CHR$
(253);:GOTO 2400
2450 SECTOR=SECT:GOSUB 10000:GOTO 100
2500 ? "TYPE * TO LOAD SECTOR ";SECTOR
;" ";
2520 INPUT SECTOR$:IF SECTOR$<>"*" THE
N 100
2530 ? :? "Ok. -- LOADing Now....";
2550 GOSUB 10000:GOTO 100
3000 REM <<<<<<< Save Sector >>>>>>>
3100 TRAP 40000:CMD=1:GRAPHICS 0:POKE
709,15:PRINT
3110 ? "       SAVE Sector Routine.":?

3200 ? " Which Sector <HEX> --> ";
3210 INPUT SECTOR$:IF SECTOR$="" THEN
PRINT :GOTO 3400
3220 IF LEN(SECTOR$)<>3 THEN 3000
3230 I=ASC(SECTOR$(1,1)):J=ASC(SECTOR$
(2,2)):K=ASC(SECTOR$(3,3))
3240 IF I<65 THEN I=I-48:GOTO 3260
3250 I=I-55
3260 IF J<65 THEN J=J-48:GOTO 3280
3270 J=J-55
3280 IF K<65 THEN K=K-48:GOTO 3300
3290 K=K-55
3300 SECT=I*256+J*16+K
3310 IF SECT<1 OR SECT>5720 THEN ? CHR
$(253):GOTO 3000
3320 SECTOR=SECT:GOSUB 10000:GOTO 100
3340 GOTO 100
3400 ? " Which Sector <DEC> --> ";
3410 INPUT SECTOR$:IF SECTOR$="" THEN
3500
3430 TRAP 3400:SECT=VAL(SECTOR$):TRAP
40000
3440 IF SECT<1 OR SECT>720 THEN ? CHR$
(253);:GOTO 3400
3450 SECTOR=SECT:GOSUB 10000:GOTO 100
3500 PRINT
3510 ? :? "Type * To SAVE Sector ";SEC
TOR;" ";
3520 INPUT SECTOR$:IF SECTOR$<>"*" THE
N 100
3530 ? :? "Ok. -- SAVEing Now....";:GO
SUB 10000:GOTO 100
4000 REM <<<<<<< EDIT SECTOR >>>>>>>
4040 GRAPHICS 0:POKE 709,15:POKE D,D1:
TRAP 40000
4050 ? "Sector>";:PRINT SECTOR
4055 POSITION 14,0:? "File>";
4060 BYTE=PEEK(BUFF+125):GOSUB 11000
4065 FILE=(BYTE-B(1)*2-B(0))/4:? FILE;

4070 NXTSEC=PEEK(BUFF+126)+256*(B(1)*2
+B(0))
4074 IF PEEK(BUFF+127)>127 THEN NXTSEC
=0
4075 POSITION 23,0:? "Next Sec.>";NXTS
EC
4080 POSITION 8,1:? "$";
4085 X=INT(SECTOR/256):? HEX$(X*2+2,X*
2+2);:X=SECTOR-X*256:? HEX$(X*2+1,X*2+
2);
4090 POSITION 18,1:? "$";HEX$(FILE*2+1
,FILE*2+2);
4100 POSITION 32,1:? "$";:X=B(1)*2+B(0
)
4105 ? HEX$(X*2+2,X*2+2);
4110 Y=PEEK(BUFF+126):? HEX$(Y*2+1,Y*2
+2)
4160 ? "  0  1  2  3  4  5  6  7  0123
4567";
4170 POSITION 3,3:? CHR$(17);
4180 FOR X=1 TO 8
4184 ? CHR$(18);CHR$(18);CHR$(32);
4186 NEXT X:? "--------";
4200 FOR L=0 TO 15 STEP 1
4210 POSITION 1,L+4:? L;
4220 POSITION 3,L+4:? CHR$(124);
4248 IF PEEK(C)=5 THEN 4309
```

```
4249 POSITION 4,L+4
4250 FOR PK=0 TO 7
4270 BYT=PEEK(BUFF+PK+L*8)
4280 ? HEX$(BYT*2+1,BYT*2+2);" ";
4290 NEXT PK
4300 REM ... PRINT CHARACTERS ...
4305 IF PEEK(C)=3 THEN 4400
4309 POSITION 28,L+4
4310 FOR CH=0 TO 7
4320 BYT=PEEK(BUFF+CH+L*8)
4330 IF BYT=0 THEN ? CHR$(0);:GOTO 439
0
4350 PRINT BYT$(BYT,BYT);
4390 NEXT CH
4395 IF PEEK(C)=6 THEN 4500
4400 NEXT L
4500 REM ...EDIT-IT...
4510 ? :POSITION 4,21:? "X Co-ord.->";

4516 INPUT X$:IF X$="" THEN 100
4517 IF X$="N" THEN SECTOR=NXTSEC:CMD=
0:GOSUB 10000:GOTO 4000
4518 IF X$="P" THEN GOSUB 12000:CLOSE
#1:GOTO 4000
4519 IF X$="+" THEN SECTOR=SECTOR+1:CM
D=0:GOSUB 10000:GOTO 4000
4520 IF X$="-" THEN SECTOR=SECTOR-1:CM
D=0:GOSUB 10000:GOTO 4000
4522 POSITION 4,22:? "Y Co-ord.-->";:I
NPUT Y$:IF Y$="" THEN 100
4524 TRAP 4500:X=VAL(X$):Y=VAL(Y$):TRA
P 40000
4526 IF X<0 OR X>7 OR Y<0 OR Y>15 THEN
 4510
4560 EDBYT=Y*8+X:POSITION 23,21:? "Seq
= ";EDBYT
4570 POSITION 4,22:? "
";
4580 POSITION 4,22:? "New Value <HEX>-
>";
4590 INPUT NVAL$:IF NVAL$="" THEN 4700

4600 J=ASC(NVAL$(1,1)):K=ASC(NVAL$(2,2
))
4610 IF J<65 THEN J=J-48:GOTO 4630
4620 J=J-55
4630 IF K<65 THEN K=K-48:GOTO 4660
4650 K=K-55
4660 NBYT=J*16+K
4670 ? NBYT,HEX$(NBYT*2+1,NBYT*2+2)
4680 POKE BUFF+EDBYT,NBYT:GOTO 4000
4700 POSITION 4,22:? "New Value <DEC>-
>";
4710 INPUT NVAL$:IF NVAL$="" THEN 4800

4720 TRAP 4700:NBYT=VAL(NVAL$)
4730 POKE BUFF+EDBYT,NBYT:TRAP 40000:G
OTO 4000
```

```
4800 POSITION 4,22:? "
  ";
4805 POSITION 4,22:? "New <STRING> ";
4810 INPUT NVAL$:IF NVAL$="" THEN 100
4820 FOR X=1 TO LEN(NVAL$)
4830 POKE BUFF+EDBYT+X-1,ASC(NVAL$(X,X
))
4850 NEXT X:GOTO 4000
5000 REM <<<< Examine Directory >>>>
5020 GRAPHICS 0:POKE 709,15:TRAP 40000
:CNT=0:CLOSE #1
5030 IF PEEK(C)=7 THEN OPEN #1,8,0,"E:
":POKE 709,15:P=0:POKE D,D1:GOTO 5100
5040 TRAP 100:OPEN #1,8,0,"P:":TRAP 40
000:P=1
5050 IF PEEK(C)<>7 THEN 5050
5060 P=1
5100 CNT=0:FOR I=0 TO 7:REM <<<Sectors
.>>>
5110 SECTOR=I+361:CMD=0:GOSUB 10000
5140 ? #1:? #1;"Sector ";SECTOR;" <$";

5145 X=PEEK(779)*2+2:Y=(SECTOR-INT(SEC
TOR/256)*256)*2
5150 ? #1;HEX$(X,X);HEX$(Y+1,Y+2);">
 ";
5155 ? #1;"Seq No: ";I;" <$";HEX$(I*2+
1,I*2+2);">"
5170 ? #1;"No: Flag Sects Start"
5175 ? #1;"--  ---- ----- -----"
5200 FOR J=0 TO 127 STEP 16
5205 ? #1;HEX$(CNT*2+1,CNT*2+2);"   ";

5210 X=PEEK(BUFF+J):FLAG=X
5220 ? #1;HEX$(X*2+1,X*2+2);"   ";
5230 X=PEEK(BUFF+J+2)
5240 ? #1;HEX$(X*2+2,X*2+2);
5250 X=PEEK(BUFF+J+1)
5260 ? #1;HEX$(X*2+1,X*2+2);"  ";
5270 X=PEEK(BUFF+J+4)
5280 ? #1;HEX$(X*2+2,X*2+2);
5290 X=PEEK(BUFF+J+3)
5300 ? #1;HEX$(X*2+1,X*2+2);"  ";
5390 X=0:IF FLAG>127 THEN X=1:FLAG=0
5400 FOR S=BUFF+J+5 TO BUFF+J+12
5405 Y=PEEK(S):IF X THEN IF P=0 THEN Y
=Y+128
5410 ? #1;CHR$(Y);
5450 NEXT S
5460 ? #1;".";
5470 FOR S=BUFF+J+13 TO BUFF+J+15
5480 Y=PEEK(S):IF X THEN IF P=0 THEN Y
=Y+128
5485 PRINT #1;CHR$(Y);
5490 NEXT S:IF X THEN ? #1;" ***";
5500 IF FLAG>95 THEN ? #1;" *";
5600 CNT=CNT+1:? #1:IF PEEK(C)<>7 THEN
 GOSUB 5950
```

```
5650 NEXT J:NEXT I
5800 ? #1;"    Re-run <Y> -->";
5810 INPUT X$:IF X$="Y" THEN 5000
5900 CLOSE #1:GOTO 100
5949 REM << A Little Delay! >>
5950 IF PEEK(C)=6 THEN 5950
5955 IF PEEK(C)=3 THEN POP :CLOSE #1:G
OTO 100
5960 IF PEEK(C)=6 THEN RETURN
5961 IF PEEK(C)=5 THEN POP :GOTO 5100
5965 IF PEEK(C)=7 THEN 5955
10000 REM <<<<< Disk Interface >>>>>
10100 POKE 769,UNIT
10110 IF CMD=0 THEN POKE 770,82
10120 IF CMD=1 THEN POKE 770,87
10130 POKE 772,BUFF-INT(BUFF/256)*256
10140 POKE 773,INT(BUFF/256)
10150 POKE 778,SECTOR-INT(SECTOR/256)*
256
10160 POKE 779,INT(SECTOR/256)
10300 A=USR(ADR(DISK$))
10400 IF PEEK(771)=1 THEN RETURN
10410 ? :? CHR$(253):? "ERROR --> ";PE
EK(771)
10430 ? "Type <*> To Re-try ... ";
10450 INPUT X$:IF X$="*" THEN 10300
10470 RETURN
10999 REM
11000 REM <<<<<< Bit Map Calc. >>>>>>
11100 FOR IT=0 TO 7:B(IT)=0:NEXT IT:B=
BYTE
11110 IF B>127 THEN B=B-128:B(7)=1
11120 IF B>63 THEN B=B-64:B(6)=1
11130 IF B>31 THEN B=B-32:B(5)=1
11140 IF B>15 THEN B=B-16:B(4)=1
11150 IF B>7 THEN B=B-8:B(3)=1
11160 IF B>3 THEN B=B-4:B(2)=1
11170 IF B>1 THEN B=B-2:B(1)=1
11180 IF B>0 THEN B(0)=1
11900 RETURN
12000 REM <<<<< Printer Routine >>>>>
12010 TRAP 12900:OPEN #1,8,0,"P:":TRAP
 40000
12020 FOR Y=0 TO 20:POSITION 0,Y
12030 FOR X=0 TO 39:GET #6,A:IF A<32 O
R A>122 THEN A=32
12040 PUT #1,A:NEXT X:? #1:NEXT Y:? #1
:CLOSE #1:RETURN
12900 ? CHR$(253),"ERROR ->";PEEK(851)

12910 ? "<RETURN> To Continue ..";:INP
UT OPT$:RETURN
30000 DATA 104,32,83,228,96
32767 END
```

QUICKSORT ACTION

FROM LAST ISSUE

```
; FILE:   QUICKSRT.ACT

INCLUDE"D:SORT.ACT"

; This uses the Quick Sort
; Procedures from the
; ACTION Toolkit

;***********************************
*
MODULE           ;DEFINE GLOBAL VARIABLE
S
;***********************************
*

BYTE KEY,FLAG=[0],TYP,HFLAG

CARD ARRAY S(1000)

CARD    N,K,I,T,H,
        R,N2,L,LL,HOLD,RR,J,HOLD2,JJ,R2


;***********************************
*
PROC GETKEY()             ;GETKEY ROUTINE

;***********************************
*


POKE(764,255)

DO UNTIL PEEK(764)#255 OD

KEY=PEEK(764)
POKE(764,255)

RETURN


;***********************************
*
PROC TIMER()   ;TIMER ROUTINE
;***********************************
*
```

```
DO
      PRINT("How many items do ")
      PRINT("you want to sort")
      PRINTE(" ")
      PRINTE(" ")
      PRINTE("Max = 999, Minimum = 10
")
      PRINTE(" ")
      PUT('?) N=INPUTC()
UNTIL N)9 AND N(1000
OD

OPEN(3,"D:NUM1000.RAN",4,0)
;PEN(3,"D:NUM1000.SRT",4,0)
FOR I=1 TO N
DO
       A=INPUTCD(3)
       S(I)=A
OD
CLOSE(3)


TIMER()
RETURN


;***********************************
*
PROC PRNTS()          ;PRINT RESULTS
;***********************************
*

CARD TEMP

GETKEY()

FOR I=1 TO N
DO
IF KEY=28 THEN EXIT FI
       PRINTC(I)
       PRINT(":")
       TEMP=S(I)
       PRINTCE(TEMP)
OD
RETURN
```

```
        CARD time
        BYTE p18=18,p19=19,p20=20

        flag = 1-flag
        IF flag=0        ; not 1st call
        THEN    time = p20 + 256*
                   (p19 + 256*p18)
                Print("Elapsed Time: ")

                PrintC(time/60)
                PRINT(" AND ")
                PRINTC(TIME MOD 60)
                PrintE("/60 secondS")
        FI

        p18=0   ; reset
        p19=0   ; the
        p20=0   ; clock

RETURN
;***********************************
*
PROC FIRSTSCREEN()      ;CLEAR AND PRIN
T
;***********************************
*

BYTE J,FIRST=[0]

IF FIRST=0 THEN GRAPHICS(0) FI
FIRST==+1
POKE(712,0)
J=RAND(16)      POKE(710,4+16*J)
POKE(709,12)

PRINTE(" ")
PRINTE("This is a demonstration of")
PRINTE("the Quick Sort Procedure")
PRINTE(" ")
PRINTE(" ")

RETURN


;***********************************
*
PROC INIT()     ;INITIALIZATION ROUTINE

;***********************************
*

CARD A


   FIRSTSCREEN()
```

OLD TEXT BY BYFIELD

```
10 REM ***************************
       ** OLD TEXT BY M.BYFIELD **
       ***************************
20 REM *

                                *
30 REM IF THIS IS GOING TO BE PUT INTO
       A PROGRAM THEN LIST IT TO DISK
       OR CASSETE.
40 REM THEN PUT IN A GOSUB 30000 AND
       CHANE LINE 30110 TO RETURN.

30000 REM ________________
          NEW CHARACTER SET
          ________________
30005 REM  LOAD SET FROM ROM INTO RAM

30010 CHBASE=120*256
30020 DIM CHAR$(34):RESTORE 30040
30030 FOR I=1 TO 34:READ A:CHAR$(I,I)=
CHR$(A):NEXT I
30040 DATA 104,104,133,215,104,133,214
,169,224,133,213,169,0,133,212,162,4,1
60,0,177,121,145,214,200,208
30050 DATA 249,230,213,130,215,202,208
,242,96
30060 A=USR(ADR(CHAR$),CHBASE)
30070 REM LOAD NEW CHARACTERS
30080 FOR I=0 TO 1023
30090 READ CH:POKE CHBASE+I,CH:NEXT I
30100 POKE 756,120
30110 STOP :REM CHANGE TO RETURN IF
                USED IN A PROGRAM.
30160 REM  MAIN CHARACTER DATA
30170 DATA 0,0,0,0,0,0,0,0,0
30180 DATA 24,24,24,24,0,24,0,0,102
30190 DATA 102,102,0,0,0,0,0,102,255
30200 DATA 102,102,255,102,0,24,62,96,
60
30210 DATA 6,124,24,0,0,102,108,24,48
30220 DATA 102,70,0,28,54,28,56,111,10
2
30230 DATA 59,0,0,24,24,24,0,0,0
30240 DATA 0,0,14,28,24,24,28,14,0
30250 DATA 0,112,56,24,24,56,112,0,0
30260 DATA 102,60,255,60,102,0,0,0,24
30270 DATA 24,126,24,24,0,0,0,0,0
30290 DATA 0,0,24,24,48,0,0,0,126
30300 DATA 0,0,0,0,0,0,0,0,0
30310 DATA 24,24,0,0,6,12,24,48,96
30320 DATA 64,0,0,60,102,110,118,102,6
0
30330 DATA 0,0,24,56,24,24,24,126,0
30340 DATA 0,60,102,12,24,48,126,0,0
30350 DATA 126,12,24,12,102,60,0,0,12
30360 DATA 28,60,108,126,12,0,0,126,96

30370 DATA 124,6,102,60,0,0,60,96,124
30380 DATA 102,102,60,0,0,126,6,12,24
30390 DATA 48,48,0,0,60,102,60,102,102

30410 DATA 60,0,0,60,102,62,6,12,56
30420 DATA 0,0,0,24,24,0,24,24,0
30430 DATA 0,0,24,24,0,24,24,48,6
30440 DATA 12,24,48,24,12,6,0,0,0
30450 DATA 126,0,0,126,0,0,96,48,24
30460 DATA 12,24,48,96,0,0,60,102,12
30470 DATA 24,0,24,0,0,60,102,110,110
30480 DATA 96,62,0,0,30,55,103,103,111

30490 DATA 59,0,0,30,51,115,126,115,12
7
30500 DATA 0,0,30,51,96,96,112,63,0
30510 DATA 0,60,102,99,99,99,126,0,0
30530 DATA 30,51,96,124,96,127,0,0,30
30540 DATA 51,96,124,96,96,0,0,30,51
30550 DATA 96,110,99,62,0,0,99,99,99
30560 DATA 127,99,99,0,0,127,24,24,24
30570 DATA 24,127,0,0,3,3,3,115,54
30580 DATA 60,0,0,103,110,124,124,108,
111
30590 DATA 0,0,112,112,96,96,99,127,0
30600 DATA 0,99,99,119,127,107,99,0,0
30610 DATA 124,118,118,118,118,119,0,0
,28
30620 DATA 54,99,99,54,28,0,0,30,51
30630 DATA 51,62,48,48,0,0,28,54,99
30650 DATA 99,111,62,3,0,60,54,54,62
30660 DATA 51,51,0,0,30,51,96,62,3
30670 DATA 127,0,0,63,108,108,12,12,27

30680 DATA 0,0,51,51,51,51,99,62,0
30690 DATA 0,99,99,99,54,60,24,0,0
30700 DATA 99,99,107,127,119,99,0,0,99

30710 DATA 102,60,28,54,99,0,0,99,99
30720 DATA 54,30,12,24,0,0,63,102,12
30730 DATA 24,51,126,0,0,30,24,24,24
30740 DATA 24,30,0,0,64,96,48,24,12
30750 DATA 6,0,0,120,24,24,24,24,120
30770 DATA 0,0,8,28,54,99,0,0,0
30780 DATA 0,0,0,0,0,0,255,0,0
30790 DATA 54,127,127,62,28,8,0,24,24
30800 DATA 24,31,31,24,24,24,3,3,3
30810 DATA 3,3,3,3,3,24,24,24,248
30820 DATA 248,0,0,0,24,24,24,248,248
30830 DATA 24,24,24,0,0,0,248,248,24
30840 DATA 24,24,3,7,14,28,56,112,224
30850 DATA 192,192,224,112,56,28,14,7,
3
30860 DATA 1,3,7,15,31,63,127,255,0
30870 DATA 0,0,0,15,15,15,15,128,192
30890 DATA 224,240,248,252,254,255,2,4
2,0
30900 DATA 21,21,0,42,2,32,170,0,84
30910 DATA 84,0,170,160,255,255,0,0,0
30920 DATA 0,0,0,0,0,0,0,0,0
30930 DATA 255,255,0,0,0,0,240,240,240

30940 DATA 240,0,28,28,119,119,8,28,0
30950 DATA 0,0,0,31,31,24,24,24,0
30960 DATA 0,0,255,255,0,0,0,24,24
30970 DATA 24,255,255,24,24,24,0,0,168

30980 DATA 84,42,84,168,0,213,85,255,1
70
30990 DATA 170,255,85,93,192,192,192,1
92,192
31010 DATA 192,192,192,0,0,0,255,255,2
4
31020 DATA 24,24,24,24,24,255,255,0,0
31030 DATA 0,240,240,240,240,240,240,2
40,240
31040 DATA 24,24,24,31,31,0,0,0,120
31050 DATA 96,120,96,126,24,30,0,0,24
31060 DATA 60,126,24,24,24,0,0,24,24
31070 DATA 24,126,60,24,0,0,24,48,126
31080 DATA 48,24,0,0,0,24,12,126,12
31090 DATA 24,0,0,0,24,60,126,126,60
31100 DATA 24,0,0,0,30,54,118,118,159
31110 DATA 0,0,96,96,124,102,102,253,0

31130 DATA 0,0,60,112,96,112,223,0,0
31140 DATA 6,6,62,102,102,255,0,0,0
31150 DATA 28,54,54,28,247,0,0,28,56
31160 DATA 48,62,123,217,28,0,0,30,58
31170 DATA 114,222,135,60,0,96,96,96,1
24
31180 DATA 102,231,0,0,24,0,24,24,126
31190 DATA 195,0,0,28,0,28,60,111,205
31200 DATA 60,0,48,48,62,54,60,247,0
31210 DATA 0,24,24,24,24,24,231,0,0
31220 DATA 0,51,127,127,219,219,0,0,0
31230 DATA 124,102,102,102,231,0,0,0,6
0
31250 DATA 102,103,231,60,0,0,0,124,10
2
31260 DATA 102,124,231,96,0,0,62,102,1
02
31270 DATA 126,143,14,0,0,96,126,102,1
02
31280 DATA 195,0,0,0,14,27,51,99,223
31290 DATA 0,0,12,63,12,12,28,247,0
31300 DATA 0,0,115,51,51,51,223,0,0
31310 DATA 0,99,54,54,127,205,0,0,0
31320 DATA 99,99,107,127,221,0,0,0,51
31330 DATA 126,110,219,177,0,0,0,27,27

31340 DATA 31,54,207,24,0,0,0,102,91
31350 DATA 219,131,62,0,24,60,126,126,
24
31370 DATA 60,0,24,24,24,24,24,24,24
31380 DATA 24,0,0,0,24,24,0,0,0
31390 DATA 8,24,56,120,56,24,8,0,16
31400 DATA 24,28,30,28,24,16,0
```

PATCH 2.5

```
10 REM Modification to DOS 2.5 to
11 REM store DUP.SYS and MEM.SAV
12 REM in the bank switch RAM
13 REM behind the OS ROM from $C000
14 REM to $F8ff
15 REM
16 REM This mod for 64K XL's only
20 REM Adapted from ANALOG #24 by
21 REM Robert Luce
22 REM
23 REM ******************************
24 REM written by Alec Benson 6/85
30 REM from FEEDBACK ADELAIDE Atari
31 REM Club, Box 333, Norwood,
33 REM Australia S.A. 5067 Aug '85
34 REM ******************************
40 REM REPRINTED ACE Newsletter
41 REM 3662 Vine Maple, Eugene, OR
100 CK=0:DIM A$(339)
105 ? :? "Reading Data...."
110 FOR I=1 TO 339
120 READ A
130 CK=CK+A
140 A$(LEN(A$)+1)=CHR$(A)
150 NEXT I
160 IF CK<>41072 THEN ? "ERROR IN DATA
 STATEMENTS-CHECK TYPING":END
170 OPEN #1,8,0,"D:PATCH25.OBJ":PRINT
#1;A$;:CLOSE #1
180 ? :? :? "D:PATCH25.OBJ CREATED:END
"
1000 DATA 255,255,231,20,233,20,32,192

1010 DATA 23,70,23,138,23,32,85,24
1020 DATA 169,0,133,212,133,214,169,29

1030 DATA 133,215,169,192,133,213,162,
16
1040 DATA 32,119,24,169,216,133,213,16
2
1050 DATA 7,32,119,24,32,70,24,96
1060 DATA 169,0,133,212,169,224,133,21
3
1070 DATA 160,0,162,3,177,212,72,32
1080 DATA 85,24,104,145,212,32,70,24
1090 DATA 200,208,241,230,213,202,16,2
36
1100 DATA 96,234,182,23,0,24,240,73
1110 DATA 32,70,23,206,158,23,48,65
1120 DATA 32,108,21,32,105,23,169,255
1130 DATA 141,158,21,141,157,21,162,16

1140 DATA 169,47,157,68,3,169,24,157
1150 DATA 69,3,32,164,21,32,85,24
1160 DATA 162,21,169,0,133,212,133,214
1170 DATA 169,31,133,215,169,228,133,2
13
1180 DATA 32,119,24,32,70,24,169,0
1190 DATA 141,157,21,141,158,21,76,146

1200 DATA 25,19,24,39,24,32,85,24
1210 DATA 169,0,133,214,133,212,169,22
8
1220 DATA 133,215,169,31,133,213,162,2
1
1230 DATA 208,18,58,24,146,24,32,119
1240 DATA 24,32,70,24,206,157,21,76
1250 DATA 152,32,32,102,24,88,169,112
1260 DATA 141,14,212,169,10,141,14,210

1270 DATA 96,120,169,0,141,14,212,141
1280 DATA 14,210,173,1,211,41,254,76
1290 DATA 107,24,173,1,211,9,1,141
1300 DATA 1,211,96,234,234,234,234,32
1310 DATA 156,25,96,160,0,177,214,145
1320 DATA 212,200,208,249,230,213,230,
215
1330 DATA 202,208,242,96,234,234,234,2
34
1340 DATA 234,234,234,234,234,234,234,
63
1350 DATA 25,109,25,32,85,24,169,0
1360 DATA 133,212,133,214,169,29,133,2
13
1370 DATA 169,192,133,215,162,16,32,11
9
1380 DATA 24,169,216,133,215,162,7,32
1390 DATA 119,24,32,70,24,96,234,234
1400 DATA 234,234,234,234,234,234,234,
234
1410 DATA 234,234,49,31,53,31,178,174
1420 DATA 181,216,204
```

TONEDIAL

```
10 REM **** TONE DIAL
80 DIM F1(11),F2(11),C1(11),C2(11),PN$
(20)
90 FOR X=0 TO 3:SOUND X,0,0,0:NEXT X:P
OKE 53768,120
100 FOR X=0 TO 11:READ A,B,D,E:F1(X)=A
:C1(X)=B:F2(X)=D:C2(X)=E:NEXT X
120 ? "ENTER #"
130 INPUT PN$:TRAP 120
150 FOR X=1 TO LEN(PN$)
210 N=VAL(PN$(X,X))
230 POKE 53762,C1(N):POKE 53760,F1(N):
POKE 53766,C2(N):POKE 53764,F2(N)
240 POKE 53767,168:POKE 53763,168
250 FOR D=1 TO 16:NEXT D
270 POKE 53767,0:POKE 53763,0
290 FOR D=1 TO 8:NEXT D
300 NEXT X
330 GOTO 120
350 DATA 150,2,176,3
360 DATA 221,2,253,4
370 DATA 150,2,253,4
380 DATA 87,2,253,4
390 DATA 221,2,131,4
400 DATA 150,2,131,4
410 DATA 87,2,131,4
420 DATA 221,2,19,4
430 DATA 150,2,19,4
440 DATA 87,2,19,4
450 DATA 221,2,176,3
460 DATA 87,2,176,3
```

ACTION

SORT

CON'T

```
;************************************
*
PROC MAIN()
;************************************
*


DO
INIT()

CSORT(S,N,0)
TIMER()

POKE(752,1)
PRINTE(" ")
PRINTE("   Press any key to see list")

printe("        Press ESC to exit")
PRNTS()
IF KEY=28 THEN EXIT FI



OD

RETURN
```

SECTOR

The Missing Link

(reprint: Page 6)

The idea of SECTOR is to allow those of you with an Atari 810 disk drive to experiment without being limited by DOS to the file structure. With SECTOR you are able to load, edit, and save ANY sector on the disk. With a reasonable understanding of DOS II's file structure you can perform all kinds of "nifty" things, such as retrieving deleted files and repairing damaged files. Examining and altering auto-boot disks is also greatly simplified. As an understanding of the way data is stored by DOS II will be of help, I will briefly outline its file structure.

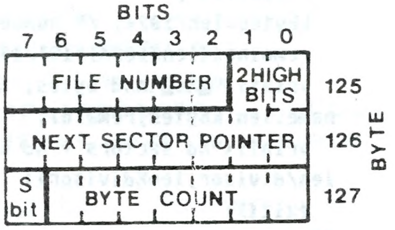
The 810 drive organizes the floppy disk as a collection of numbered blocks of bytes called "sectors". There are 720 sectors or blocks, and each holds 128 bytes or characters. As each file is created, an empty block is found and the data is poured into it. When the sector is filled, for another free sector has to be allocated and somehow linked to the first so when the file is being read the second sector can be found.

The directory information for this file only tells the DOS where to find its first sector. So how does DOS find the rest of the file? Well, only 125 bytes in each block are used for the user's data. The remaining 3 bytes are kept and used by DOS to provide 3 functions: 1. To point to the next sector in the file; 2. To say which number file the sector belongs to; and 3. To indicate if the sector is a "short" sector, and if so how many bytes are valid.

The pointer is obviously the key to the way in which DOS finds the next sector allocated to the file. The second function is not really essential, but it is useful, because as DOS created the file it notes the occurrence of the file's name entry in the directory and places this number into one of the last 3 bytes of each sector used by the file. Whenever the file is read back, if there is ever a discrepancy between the value of this byte and the directory, DOS assumes there has been some problem. It will report this to the user as the dreaded "ERROR 164", file number mismatch.

This unhappy event is usually caused by the careless user either "BREAK"ing or "SYSTEM REST"ing during a disk operation, or swapping disks in a drive while a file is still open on the drive. Both are !!**!!* mistakes which should be avoided at all costs!

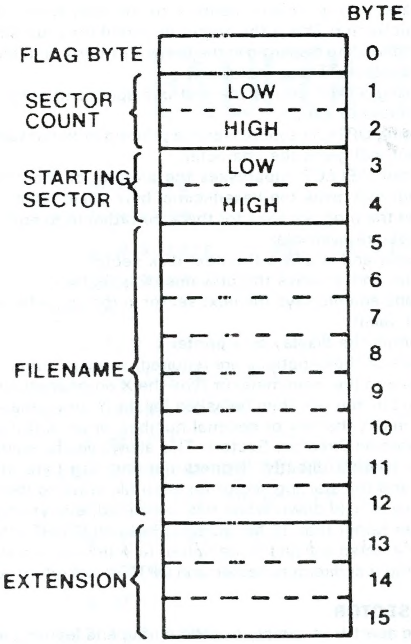
The last function is a vital one, for a file may not have used all the bytes in its last sector, and if this is the case DOS needs to know this fact and how many bytes of that sector are allocated to the file. There are 3 functions and 3 bytes, so it seems logical to have one byte per function. This cannot be so, however, because there are 720 sectors on a disk. So more than one byte is needed to store the next sector information. Since the directory position byte number does not have to be larger than 63, it does not require all 8 bits of its byte, so two of its bits are used by the next sector pointer.



This is how the sector's bytes are allocated. Looking at the diagram you will see one of the bits has not been explained yet, the "S" bit on byte 127. If the last sector of a file is not completely used, then the "S" bit is set to logic high, and the BYTE COUNT will give the actual number of valid bytes.

THE DIRECTORY

There are 8 sectors (361-368) allocated to the disk directory, each holding 8 entries; i.e., 64 entries total. The 16 bytes of each file directory are allocated as follows:



The flag byte is used to indicate the status of the file and the bits are mapped as follows:

BIT No * IF SET HIGH, THEN	
BIT 7 *	FILE HAS BEEN DELETED
BIT 6 *	FILE ENTRY EXISTS
BIT 5 *	FILE IS LOCKED
BIT 0 *	FILE IS OPEN FOR OUTPUT

The flag byte is used to indicate the status of the file and the bits are mapped as follows:

BIT #————	If set HIGH, then
7	File has been deleted
6	File entry exists
5	File is locked
0	File is open for output

Thus the flag byte may have the following values:

VALUE ————	STATUS
$00	Entry not yet used
$40	Entry in use (normal closed file)
$41	Entry is in use (and file is currently open for output)
$60	Entry is in use (AND file is locked)
$80	Entry is available (prior file has been deleted)

The sector count (number of sectors in the file) and the starting sector number are obvious, as is the filename. Note, however, that DOS does not insert the full stop before the extension. The directory manager routines remove and insert this for the user's convenience.

THE PROGRAM

The precise format of the program is important, so be careful to include the correct number of spaces and characters where applicable, otherwise you may find some strange numbers will result. The program, when first RUN, will take over 10 seconds to initialize its string and arrays, so if you BREAK out of the program you can resume it by typing GOTO 100 and avoid the long 10-second wait. This continues without having to re-initialize! The program is based around a menu and has 5 options.

1. Normal Directory Listing: This gives the standard disk directory listing in two column format. Typing RETURN will take you back to the menu. Any other key will re-run the directory.

2. Load Sector: This allows you to load a sector into the buffer, and will first ask you for the sector number in "HEX", i.e. 001 to 2D0. Typing RETURN only will cause the program to ask for a decimal value in the range 1 to 720. A further RETURN will allow you the option of loading the current sector by typing '*'.

3. Save Sector: This is identical to the load sector option in use.

4. Edit Sector: This is the major section of the program. The current buffer contents are displayed in the form of a matrix, and there are several options available. These include:

Pressing START aborts the matrix display and asks for the X and Y coordinates to edit;

Pressing OPTION suppresses the printing of the hexadecimal buffer listing, but still gives the character;

Pressing SELECT suppresses the printing of the character buffer listing, but still gives the hexadecimal byte value table.

When the program asks for the X co-ordinate to edit, the following commands are available:

+ loads and displays the next disk sector.

- loads and displays the previous disk sector.

N loads and displays the next sector in the same file as the current sector (if valid).

P dumps the display to a printer.

If none of these options are required, you can either type RETURN to get back to the main menu or type the X co-ordinate of the byte you wish to alter. You will then be asked for the Y co-ordinate, after which you can insert the hex or decimal number, or an ASCII string.

5. Examine Directory Sectors: This allows you to examine the disk's directory sectors directly. It prints out the flag byte, the number of sectors, and the starting sector for each file entry to the screen. If the START key is held down while this is entered, everything is printed to the printer rather than to the screen. Pressing START after the routine has been entered will pause the output to screen or printer. SELECT will retart at the first directory sector, and OPTION will return you to the main menu.

USING SECTOR

Quite apart from simply experimenting and learning about the disk system, there are many practical uses for SECTOR. For instance, if you have accidentally deleted an irreplaceable file on a disk, it can be retrieved by finding its old directory allocation using option 5 (examine directory sectors), then using the sector edit facility to alter the FLAG byte to $40. You should then copy off to a fresh, formatted disk all the desired files.

One important part of the ATARI DISK SYSTEM is the VOLUME TABLE OF CONTENTS (sector 360) in which DOS keeps track of which sectors are in use and which are free for new or extended files. This is the subject of another article. Meanwhile I hope you have many interesting hours of experimentation.

— Ron Levy

MEETING
WEDNESDAY
NOVEMBER 13TH
7:30 PM

SOUTH
EUGENE
HIGH

Solder two wires to the two conductors on the jack. Join the two cut wires from the speaker with the ones from the jack. Make sure you have got the right wires going where. In other words, the connection to the speaker is the same except in the middle the leads of the jack are attached to pick up the signal. You've got to protect these, so either wrap electrical tape around them or use those insulated connectors.

Now you're done with the TV set. It's time to solder two wires to the two conductors of the plug. Hey! You with the earphone in your TV, wake up! You've got to do this too! Ok, now you have two wires coming from your plug. Take these two wires and wrap them to the two wires on the side of the transformer which has two wires (The other side has 3, that's the 1000-ohm side). Once again, protect these connections.

Take the other side of the transformer, the one with 3 wires, and connect the two on the outside to the red and green wires of the phone line. You can do this anyway you want, I leave it up to you. I suggest protecting the transformer somehow. Mine is in my phone. Well, you're done!

TONE DIAL is a program which generates the tones and puts them through the speaker. However, I didn't write this. It's from (*Gasp*, Copyright!) A.N.A.L.O.G. Magazine issues 19 and 21.

Theoretically, this should work with ANY modem if you can put the subroutine into the terminal program. I know it works on the 835, I own one! So, now you can tone dial like the big modems do.

FSIZE BY RALPH WALDEN

```
/* FSIZE.C - returns size of file */

#include "getname.c"

main() $(
  char name[20];
  int iocb,len,divisor,kbytes,remain;
  fast();
  iocb=getread(name," ");
  divisor= (peek(0x1311+(name[1]-'0'))

== 2) ? 253:125;
 /* single or double density sectors *
/
  inverse(name);
  len=bgets(lomem(100),highmem()-
lomem(100),iocb);
  closeall();
  kbytes=len/1024; /* number of K */
  remain=((len%1024)*10)/1024;
  printf("█%s█\n%d bytes, %d.%dK\n",
name,len,kbytes,remain);
  printf("%d sectors + %d bytes\n\n",
len/divisor,len%divisor);
  exit();
$)

/* convert a string to inverse
   characters */
inverse(str)
char *str;
$(
  while(*str) *str++ |=128;
 /* OR the inverse bit */
$)
```

10/11/85... Following is the text of an official press release issued today by Sam Tramiel, president of Atari Corp., concerning the recent agreement between DRI and Apple.

SUNNYVALE, CALIFORNIA -- In an agreement between Digital Research and Apple Computer, Digital Research has agreed to make certain superficial changes to three of its application programs: "GEM Desktop", "GEM Paint", and "GEM Draw". Contrary to reports, the agreement does not require changes to GEM orto Atari Corp.'s TOS operatingsystem.

DRI, as part of its normal development process, has prepared enhancements to its GEM application products which further set it apart from the Apple Macintosh. Atari is reviewing these changes. Atari promises to its software developers and custmers that Atari will make no changes that will reduce the capabilities of the Atari ST system and software. Further, Atari promises that any enhancements will maintain compatability with GEM applications software on the market and in development.

Digital Research's GEM and the applications software which uses its capabilities will continue toprovide the most powerful andeasy-to-use computer interface onthe market. The Atari 520ST continues to be the most powerful computer delivering these features and will continue to do so at an affordable price.

FADEOUT

```
10 GOTO 300
97 REM ****
98 REM * FADE IN AND OUT SUBROUTINE *
99 REM ****
100 D=3
110 FOR L=0 TO 14 STEP 2
120 SETCOLOR 1,0,L
130 GOSUB DELAY
140 NEXT L
150 D=100:GOSUB DELAY:D=2
160 FOR L=14 TO 0 STEP -2
170 SETCOLOR 1,0,L
180 GOSUB DELAY
190 NEXT L
197 REM ****
198 REM * DELAY SUBROUTINE *
199 REM ****
200 POKE 20,0
210 IF PEEK(20)<D THEN 210
220 RETURN
297 REM ****
298 REM * INIT ROUTINE *
299 REM ****
300 GRAPHICS 0
310 POKE 752,1
320 FADE=100:DELAY=200
330 SETCOLOR 1,0,0
340 SETCOLOR 2,0,0
347 REM ****
348 REM * MAIN LOOP *
349 REM ****
350 ? CHR$(125)
360 POSITION 10,8
370 ? "THIS IS AN EXAMPLE"
380 POSITION 17,10
390 ? "OF"
400 POSITION 13,12
410 ? "FADING TEXT"
420 GOSUB FADE
430 D=20:GOSUB DELAY
440 ? CHR$(125)
450 POSITION 6,10
460 ? "THE LETTERS FADE IN AND OUT"
470 GOSUB FADE
480 D=20:GOSUB DELAY
490 GOTO 350
```

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to Atari Corp. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are $14 a year for U.S., and $24 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.

Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

**President— Dick Barkley, 2907 Wingate, Eugene, OR 97405
503-344-5843

Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, Or 97405
503-686-1490

8-bit Librarian— Chuck & Jody Ross, 2222 Ironwood, Eugene 97401
(503) 342-4133

ST Librarian— Jim Bumpas, 4405 Dillard Rd., Eugene OR 97405
503-484-4746

Editors— Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405
503-344-6193
Jim Bumpas, 4405 Dillard Rd., Eugene, OR 97405
503-484-4746

E.R.A.C.E. (Education SIG Editor)— Nora Young, 105 Hansen Lane,
Eugene, OR 97404 / 503-688-1458

Send 50c stamps or coin ($1 overseas) to the Ness' for the new, updated ACE Library List—new in May 85!

Bulletin Board
(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of an 800 XL, 2 double-density double sided disk drives and 2 double-sided, double-density, 80-track disk drives, an Epson MX80 printer, a 1200 baud Prentice P212ST modem, running the Mindlink Bulletin Board software distributed by SofMark.

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only $12 each ($2 extra for Airmail). Available only from:
George Suetsugu
45-602 Apuapu St
Kanoehe, HI 96744

SortFinder Index

A composite index of Atari related articles in four popular computer periodicals, including ACE. Volume 1covers April, 1981 to June, 1983. Volume 2 covers July, 1983 to December, 1985. Only $6 per printed copy or $11 per disk copy for ACE members:
Jim Carr 2660 S.W. DeArmond
Corvallis, OR 97333

Atari
Computer
Enthusiasts
3662 Vine Maple Dr. Eugene OR 97405

FIRST
CLASS
MAIL